Carnegie Mellon
Software Engineering Institute

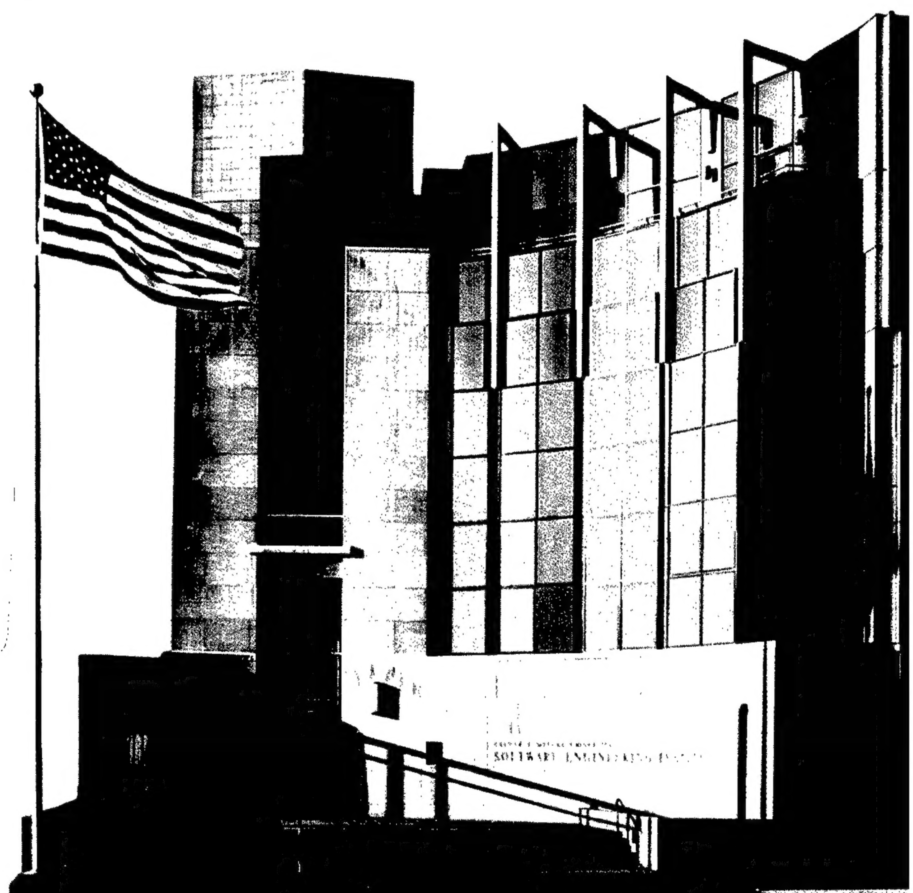
Predicting When Product Line Investment Pays

Sholom Cohen

July 2003

Product Line Practice Initiative

Technical Note
CMU/SEI-2003-TN-017



Predicting When Product Line Investment Pays

Sholom Cohen

July 2003

Product Line Practice Initiative

Technical Note
CMU/SEI-2003-TN-017

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

List of Figures

List of Tables

Acknowledgements

This document has been adapted from analyses done with Software Engineering Institute customers. The author extends his thanks for their participation in generating the results.

Abstract

A product line approach may appear very attractive, with obvious benefits in speedier time to market and higher quality, however many organizations demand financial justification before proceeding. Without knowing costs, the decision makers won't budget funds or personnel to carry out the up-front asset construction tasks. In addition, not all organizations are ready to commit up front to a full asset set, one that covers most if not all product line features. Many managers favor an incremental approach to product line adoption, one that first tackles areas of highest and most readily available commonality, earning payback early in the adoption cycle.

This report defines key factors to consider in taking an incremental approach to fielding a product line. An organization building a business case can apply these factors to show that product line investment can result in product development savings. The example presented here shows a net savings of almost $180 million in projects that would have cost about $600 million under traditional development approaches. The $180 million in savings takes into account an investment of $54 million in product line start-up costs. The example also illustrates ways to present the data needed to make a compelling business case.

1 Introduction

An organization makes a sizeable commitment of resources when it turns from single-system approaches to a product line approach [Reifer 97]. Occasionally, if an organization is faced with a desperate situation (e.g., commit to a product line approach or fail to meet customer demands), the organization must commit the resources because it simply cannot continue to produce systems one at a time [Brownsword 96, Dager 00]. More often, the organization has a number of options for upgrading from a single-system approach: flexible architecture, framework evolution, component strategy, product line, or others [Bosch 02].

A product line approach is often viewed as the most risky of these alternatives. Objections include

- a long lead time to develop assets
- an insufficient number of systems that could potentially use the assets
- a drain on critical resources

However, there are many potential benefits to product line approaches including financial, quality, and non-tangible benefits (e.g., developer satisfaction) [Clements 02]. Generally, the business case must be made either in terms of reduced development costs or reduced time to market. Given so many variables, how do you determine when product line investment pays?

This report presents an approach for making the investment determination. The approach is based on the "ABCs" of a business case.

- Applications - the different systems the organization plans to deliver using product line assets over the time period of the business case
- Benefits - the projected cost savings or other return the use of the product line assets should provide
- Costs - the actual costs of reuse the organization incurs in developing and using the assets

The projected return on investment in asset development compares the estimated costs of traditional development with the estimated costs of using assets to produce the same systems. Ideally, an organization will compare these projections to currently available empirical data and refine the estimates.

This report introduces the ABC approach using a case study of product line adoption at the U.S. National Reconnaissance Office (NRO) [Clements 01] and how the NRO built a business case to justify continued product line investment [Bergey 01]. The case study

follows an incremental introduction of assets for building applications in the product line. The incremental approach identifies those areas within the scope of the product line most amenable to development based on assets. Factors in this refinement of scope include degree of commonality, knowledge of relevant domains, stability within the domains, and preexisting assets. The NRO built a set of assets within a limited scope of a product line for ground-based command and control of satellites, and has used them in the development of operational systems.

In the first increment, the NRO developed assets with a degree of reuse (DOR) of 25%, meaning that assets were used in the development of 25% of the software of a typical product. The business case examined a second increment to reduce the costs of reuse (COR) and a third increment to extend the DOR to 50%. The case study was based on this incremental introduction of assets for building other applications in the product line. Details of the estimated DOR appear in Section 3 of this report.

These analyses demonstrate the need for thorough tracking of cost information. While they are based on actual results, they include long-term projections that will be strengthened by future product line application. These projections show results over a five-year period with two systems per year. It is also possible that more than two systems will be under development per year.

While this report presents a case study that proceeds incrementally, the ABC approach described here can also be used when the DOR is 100%; that is, where the assets cover any product in the product line. Section 2 describes the ABC approach, while contrasting it with some other business case approaches, and highlights the ability of the ABC approach to deal with incremental or complete coverage through the use of degree and cost of reuse. Section 3 presents the case study and explains the derivation of the COR, DOR, and other parameters for the NRO business case. Section 4 offers practical suggestions for presenting a business case including savings projections derived from the cost data.

2 The ABC Approach for a Product Line Business Case

The ABC approach looks at the applications, benefits, and costs of a product line. This section considers in detail what these three terms mean and how they contribute to a business case.

2.1 Applications

Several authors have discussed economic issues relating to reuse [Lim 98, Poulin 97, Reifer 97, Weiss 99, Clements 02]. They generally tout the lower development costs and higher quality of software developed from assets. But in making the business case, their economic considerations are generally based on the following assumption: an organization develops assets for a domain or for a product line, and reduces the costs of developing software applications that would otherwise be produced in single-system fashion. They do not take into consideration the time period over which the organization will apply the assets to specific applications. In some of the earlier works, the authors do not consider the broader issues of assets and product lines. Instead, they concentrate on the reuse of code components rather than the managed use of assets through a product line architecture.

For example, suppose a company is producing cell phones or other consumer electronics. In that case many products are developed per year using existing assets. The assets usually change very little from product to product. Over two or three years, however, the asset base may completely change to reflect new technology or consumer demands. This is the "refresh rate" (the assets must be refreshed once every three years).

Products that are fielded far less frequently (e.g., satellite ground control systems, large-scale medical diagnostics systems) may be adversely affected by the refresh rate. If only one product is fielded per year and the refresh rate is three years, the organization must rebuild the asset base almost as frequently as it builds products. Weiss' rule of thumb that asset payback occurs after three or four applications, or Clements' rule that "it takes two" will not apply if the refresh rate equals the time it takes to field three or four applications in the product line.

The number of applications (the "A" in the ABC method) and the time over which they are produced is, therefore, an important consideration within the ABC method. In developing a business case, the organization must make reasonable estimates about the number of applications it will field per year and the degree of change within the product line over a multi-year period. The organization may use historical data for these estimates, but will, likely, hedge this number against other business goals, such as capturing increased market

share or expanding market coverage. However, there is a law of diminishing returns here. If the ability to increase variety does not increase profit, why invest in the ability to offer greater variety? All of these considerations must be addressed in the business case.

2.2 Benefits

What about benefits, the "B" in the ABC approach? Benefits come in two categories.

1. tangible – benefits that can be measured directly, such as reduced time to market or reduction in defect reports
2. intangible – benefits that developers report but cannot measure in terms of product metrics. These benefits may include developer satisfaction or customer acceptance. In some cases, intangible benefits may be measured indirectly in terms of developer turnover or repeat customers, though many factors beside systematic reuse play a role in those results.

Early in planning for a product line, an organization should set business goals and define the benefits it hopes to achieve through the product line approach. The business case presented here looks at benefits in terms of productivity. Table 1 provides a list of possible tangible benefits realized from product line adoption [Clements 01].

Table 1: Tangible Benefits from the Product Line Approach

Factor	Benefits to Organization Using a Product Line Approach
Profitability	The asset base allows the organization to produce products targeted to specific market segments. The benefit of this targeting is seen in increased market share and the overall profitability of the organization.
Quality	A reduction in the number of defect reports is typical for a system of this type. Quality may also be measured in terms of the time to repair and the ripple effect: are fixes handled locally with no ripple effects and no effect upon the architecture, or do fixes require extensive redesign?
Performance	Use of assets improves performance over results predicted without assets. In places where reuse may lead to timing problems, the assets provide variation points that may be used to circumvent software assets and apply faster algorithms.
Integration time	Incremental builds were completed faster than non-asset portions or an application. This may be a direct carry-over from the incremental approach to development.
Code volume	The number of design objects for subsystems using the assets is lower than estimated for the single-system approach, with a similar reduction in actual source code size.
Productivity	A smaller development staff is required. The overall costs are cut by some measurable amount. The overall schedule is cut (quick time to market or to field).

Table 1: Tangible Benefits from the Product Line Approach (cont'd.)

Factor	Benefits to Organization Using a Product Line Approach
Productivity (cont'd.)	There is documented flexibility in meeting customers' requests for modifications. Assets are treated like a commercial off-the-shelf (COTS) product (initial training is required, then development proceeds based on domain specification, interface definitions, and documented asset usage guides).

Table 2 provides a similar list for intangible benefits.

Table 2: Intangible Benefits from the Product Line Approach

Factor	Benefits to Organization Using a Product Line Approach
Attrition rate	Lower staff turnover occurs after product line adoption compared to other projects not using a product line approach.
Developer acceptance	After initial training, developers report satisfaction with the asset-based approach and with the architecture.
Professional satisfaction	Developers report that the pedestrian tasks of coding have already been done (in the software assets); they can focus on more interesting, mission-specific capabilities or on performance tuning.
Customer satisfaction	Assets reduce risk by improving the predictability of delivery and lower defect rates. Customers return for repeat business based on the product line approach.

2.3 Costs

One factor that is often not considered is the cost (the "C" in the ABC approach) of *using* assets. While the methods advanced by Weiss and others allow for the cost of developing assets, there may be some, often considerable, cost in applying those assets in the development of products. This cost of reuse may vary from 0% (where there is no effort to use assets to produce the end product) to 100% (where the cost to reuse assets equals the cost of developing the end product software from scratch). The COR may even exceed 100%; some contend that software reuse costs more than it is worth when anything beyond trivial tailoring of assets is required [Glass 03].

Costs of reuse include

- developing requirements and design to a point where assets may be used[1]
- determining which assets to use and how to use them

[1] Where the asset base supports requirements and traceability to architecture and design, this cost may be very low in terms of a complete application (e.g., constructors).

- testing assets for suitability, and integrating the assets and non-asset-based elements of an application
- reporting results of asset use back to asset developers
- training to familiarize developers with core assets including the architecture, components, and test facilities

For sensitivity analysis of the cost projections, the business case may use reuse costs of 10%, 25%, and 40% to offer a range of possible values. The analysis in the NRO business case includes only the 10% and 25% COR.

Another factor to consider in making the business case is the portion of a complete product that is supported by the assets (the degree of reuse). Weiss' model shows that product line investment pays after three or four systems [Weiss 99, pp. 45-49]. The organization recovers the costs of developing and maintaining the assets as the costs of product development come down. However, this model and others make assumptions about the custom-built portions of each system. For instance, a generator or comprehensive asset base is used to construct the systems in Weiss' model. The custom-built portion is small compared to what is being generated or produced from component assets. With limited coverage of assets, where the organization builds only a fraction of the final system from assets, an organization must consider the degree of reuse offered by the asset base.

Development, use, and maintenance of a production plan can also bring down the COR [Chastek 02]. The production plan gives developers detailed guidance in the use of assets for developing products in a product line. The plan assures the correct usage of assets and defines a process for reflecting asset use back into the product line for continuous improvement. Without such a plan or some defined (and maintained) means of asset use, the asset base may diminish in value to the level of a component repository, with higher costs of application to the product line.

The refresh rate, the point at which old assets must be retired and new assets built to accommodate changes in the product line, affects cost and the degree of reuse. If the refresh rate is short, applications are changing rapidly and the asset base must be rebuilt frequently to accommodate that change as new products are developed. If the assets are never refreshed, the DOR decreases rapidly over time and the COR increases due to necessary changes in existing assets. When the refresh rate is long, assets have longer shelf lives and undergo less change with each new product. The COR can be brought down through investment in tools, training, or processes to improve usability.

The payoff of product line investment is not guaranteed when there is a longer lifetime for assets, nor is success precluded by short refresh rates. The refresh rate is merely a parameter that must be measured in building the business case.

2.4 The ABC Approach and Incremental Product Line Development

What if an organization wants to develop assets and field a product line in an incremental fashion? Under this approach, the organization plans from the beginning to develop a product line. It develops part of the core asset base, including the architecture and some of the components, and then develops one or more products. In the next increment, it develops a portion of the rest of the core asset base and develops additional products. Over time, it evolves more of the core asset base in parallel with new product development.

The relevant issues to address in assessing the merits of this approach include

- Asset coverage, measured as the DOR, starts at far less than 100% of any one application, and may increase in planned increments. What are the costs of such a strategy?
- What cost data are required and how should they be presented?
- What benefits are gained?
- When does the product line investment pay off under this strategy?

If the organization wants early feedback, an incremental approach offers certain benefits. The organization can cancel use of the asset base on a specific product or cancel the entire product line approach based on these early results.

The ABC approach offers guidance specific to the problem of incremental adoption. It can show the benefits to be gained with less up-front commitment of funds, flexibility in allocating time resources between asset and product development, and a shorter lead time to produce the first product. (This was the situation desired by the NRO.) A key component of the decision process is measuring the DOR. While the size of the assets used (in lines of code, function points, or some other measurement of a total product) accounts for part of the DOR measure, one must also account for the value of a software architecture or other assets for the overall product. Even where software component assets offer only partial coverage for populating the entire structure, the architecture or test assets may apply across an entire product.

To illustrate incremental adoption, let's suppose an application produced in the product line requires 100K lines of code. Assume that the software component assets in the organization's asset base will account for 20K, but the architecture defines the structure of the entire 100K, even for that portion of the product for which there are no component assets. The developers are still responsible for some portion of the overall requirements and testing for the entire product, but having requirements, product architecture, test, and other assets may account for 10-20% of the overall software development. The DOR will be at least 20% when considering both component and non-component assets. Boosting the DOR to 30% to account for the architecture and other assets may be a reasonable first guess, later backed up by actual data as the first products in the product line are delivered.

Now the organization wants to consider whether to build new assets to increase the DOR to 50%. The ABC approach looks at applications where the organization will apply the enhanced asset base, the benefits gained by developing those products from the enhanced asset base, and costs of the enhancement. The NRO took this approach to fielding its product line. The next section examines the entire business case built by the NRO.

3 Business Case for an Incremental Approach

Several key factors influence the cost projections for applying a product line approach. Using the ABC approach, these factors may be summarized as

- Applications: how many systems will be fielded over a time period and how much will they vary over that time period?
- Benefits: what are the goals the organization wishes to achieve in adopting a product line approach? How does continued investment in assets help meet these goals?
- Costs: what are the historical costs for products built in the product line and how will these costs be affected by the development and use of an asset base?

To establish when product line investment pays, an organization must compare the current, single-system approach to the product line approach. In the NRO business case, the organization also considered incremental introduction of their product line strategy.

- Increment 1. Build and maintain baseline assets to cover a subset of system features and use those assets as the basis for future development. Maintenance includes fixes in response to defect reports and improvements to performance, usability, or other quality attributes.
- Increment 2. Commit funds for further investment in refinements to baseline assets. Maintenance activities go beyond those of Increment 1 to include support for training in use of the assets, new tools, and process improvement to support better feedback of results. This investment lowers the COR.
- Increment 3. Expand feature and asset coverage into new domains not currently addressed by existing assets. This investment increases the DOR.

From the initial increment, the NRO established and maintained a software product line architecture as the basis for asset and product development. This architecture would evolve, especially in Increment 3, due to the addition of new components to address new domains.

3.1 Applications

The NRO built its business case on the assumption that there would be, on average, two new systems per year in the product line. They assumed that the characteristics of new systems would be similar to those of past systems, and their incremental strategy to product line adoption would, in the first increment, address those areas of the product line that change little, if at all, from application to application. The business case selected a DOR of 25% for the first increment, meaning that 25% of a new system may be derived from existing assets.

This DOR was based on the results obtained from the first use of the assets in producing a member of the product line and was assumed to apply uniformly in the near term across future systems. During the second increment, the organization chose to invest in improved asset usability. The business case shows the organization investing in improved tools and training, lowering the overall COR. Finally, the business case proposed investment to expand the scope of the assets to increase the DOR to 50% during the third increment; however, the refresh rate of the newer set of assets will be shorter than that of the first set. This fact will be further explored under costs of reuse in Section 3.3.

3.2 Benefits

The NRO set specific goals that must be achieved under the product line approach. As a government entity, the NRO is not concerned with profitability, but rather with cost avoidance. It wanted to reduce the costs of development by at least 20% to make the reuse investment visible to the organization. Defect reports were to be reduced by more than half, and, under the product line approach, there was to be no adverse effect on software performance. The NRO wanted to see a 50% reduction in integration times, since they were major cost drivers. Similarly, the organization wanted to see reductions in time to field, since delays in fielding carry major cost increases. As the product line was institutionalized, the NRO wanted to retain developers from system to system and hoped the product line approach would attract new, skilled workers; a growing development organization expands the ability to develop more systems within the product line.

3.3 Costs

The costs section of the business case must use actual data plus projections to illustrate the benefits of adopting a product line approach. The costs include

1. historical costs – the costs to develop systems one at time without assets

2. asset development costs – the costs of developing assets and sustaining them during the period covered by the business case

3. increment costs and cost savings – the costs of implementing each of the three increments and the projected savings achieved by each

3.3.1 Historical Costs

The NRO business case looked at data from legacy developments to determine at what point the product line investment would pay. The organization used historical costs from developing systems in single-system fashion for making this determination. It selected systems comparable to those it would build in the future under a product line approach. Table

3 illustrates the historical costs of developing systems using a single-system approach. (Note: the dollar amounts used in the following tables are based on actual organization results for legacy costs.)

Table 3: Historical Costs of Systems

Program Name	Lines of Code	Development Cost (in millions)	Maintenance Cost per Year (in millions)	Number of Years in Operation	Cost with Maintenance (in millions)
AA	122500	17	1.70	4.00	23.80
BB	300000	46	4.60	3.00	59.80
CC	286000	31	4.10	2.00	39.20
DD	640000	63	9.70	1.00	72.70
EE	500000	118	15.00* * - budgeted	0.00	118.00

The table includes maintenance costs to show the true life-cycle costs for each system. The figures do not include any projections for net present value or other "cost of money" figures. These figures could be included in the analysis for greater accuracy in determining the true costs of the product line investment.

3.3.2 Asset Development Costs

The actual asset base cost $16 million to develop. The asset base must also be sustained over time as assets are used in the product line. Sustainment addresses defect reports and incorporates feedback from users. The organization incurs sustainment costs as the asset base is used in successive years after development. These sustainment costs occur in three main categories.

1. routine maintenance, including defect repair and limited perfective enhancements of existing assets based on user feedback
2. development of new assets, extending the asset base with assets in existing domains (Increments 1 and Increment 2) or in new domains (Increment 3)
3. improved packaging of assets to support new programs, improving the ability of users to make use of assets (Increments 2 and 3)

The activities that may be considered "routine" maintenance are covered by the $1.6 million per year figure, 10% of the development costs using normal maintenance rates.[2] Sustainment costs of individual products are already included under historical costs in Table 3.

3.3.3 Increment Costs and Cost Savings

The cost projections in the business case assumed that, where assets were available, new systems would use those assets, but that no one system could make use of all the assets. So although the total lines of code (LOC) in the asset base may exceed 150K, a 500K system will use less than 100K of the total LOC. The NRO obtained this result from the first member of the product line. (See Section 3.1.) The DOR for calculations in the business case is set at 25%, increased slightly from the ratio of asset code (100K) to total system code (500K) to account for the effects of using a product line architecture and other product line assets. (See Section 2.4.)

The costs of the three product line increments cover the use and sustainment of baseline assets, enhanced sustainment, and building new assets. While there will be savings from cost avoidance for future systems under each increment, these savings must be reduced by the development and sustainment costs of the assets. The following assumptions were used.

- two systems per year over five years, labeled System 1 through System 10, based on size characteristics of the legacy systems in Table 4. (For this analysis, Systems 1 and 2 correspond to Program AA in Table 3, Systems 3 and 4 to Program BB, etc.)
- use existing assets under Increment 1
- DOR = 25%
- COR = 25%

Thus, the total cost avoidance for Increment 1 will be over $88 million. Cost avoidance shows the following relationship:

cost avoidance = cost without reuse – cost with reuse – reuse expenditures (COR)

Table 4 shows results of the analysis for Increment 1 across 10 systems. Cost with reuse takes actual costs from Table 3. Cost with reuse shows cost avoidance (based on a DOR of 25%) by use of the asset base reduced by the COR (25%). For example, System 1 costs were reduced by $4.46 million, including 25% of $23.8 million ($5.95 million) minus the $1.49 million cost of reuse (25% of the $5.95 million savings). Table 4 also shows the cumulated costs of building and sustaining assets under "Cumulated reuse expenditures."

[2] The data used in formulating the historical costs show actual maintenance as 10-15% of development costs per year. Similarly, the data used in formulating the Constructive Cost Model (COCOMO) parameter for maintenance—annual change traffic (ACT)—range between 5 and 20% [Boehm 81].

Table 4: Cost Savings Under Increment 1 (Five-Year Period)

System	Cost without reuse (in millions)	Cost with reuse (in millions)	Cumulated reuse expenditures (in millions)	Cumulated savings over five years (in millions)
1	23.80	19.34	16.00	-11.54
2	46.75	37.98	16.80	-8.03
3	106.55	86.57	17.60	2.38
4	164.05	133.29	18.40	12.36
5	203.25	165.14	19.20	18.91
6	240.40	195.33	20.00	25.08
7	313.10	255.89	20.80	36.41
8	380.95	311.17	21.60	47.58
9	498.95	407.65	22.40	68.90
10	609.45	497.43	23.20	88.82

Under Increment 2, the NRO planned investments to improve the usability of the existing assets. The goal of this increment was to reduce the COR from 25% to 10%. Asset improvement included

- developing tools or asset enhancement (e.g., improved user documentation of the software product line architecture) to limit the costs of developing requirements and design to the point where assets may be used
- training in asset use
- improved asset test software
- streamlining feedback from asset users to asset maintainers

Although enhanced maintenance would double sustainment costs, the savings increased by $16 million (in addition to the $88 million shown in the first option). These increased savings resulted from the lower cost of reuse. Table 5 shows the details.

Table 5: Cost Savings Under Increment 2

System	Cost without reuse (in millions)	Cost with reuse (in millions)	Cumulated reuse expenditures (in millions)	Cumulated savings over five years (in millions)
1	23.80	18.45	16.00	-10.65
2	46.75	36.23	17.60	-7.08
3	106.55	82.58	19.20	4.77
4	164.05	127.14	20.80	16.11
5	203.25	157.52	22.40	23.33
6	240.40	186.31	24.00	30.09
7	313.10	244.45	25.60	43.05
8	380.95	297.94	27.20	55.81
9	498.95	389.39	28.80	80.76
10	609.45	475.02	30.40	104.03

The final analysis in the business case looked at extending the assets into new domains during the third increment. In this analysis, there was a second investment of $16 million for new assets to increase the DOR to 50%. During this increment, the COR increased because the second set of assets had a shorter refresh rate than those developed under the first increment. (See Section 2.3.) The analysis included the enhanced maintenance levels shown in Table 2. With the additional investment of $16 million in new assets, the net increase in savings is $73 million. Table 6 projects the results of implementing Increment 3. Note the increased COR after System 4, reflecting the investment in new assets and domains.

Table 6: Cumulated Savings from New Asset Investment

System	Cost without reuse (in millions)	Cost with reuse (in millions)	Cost of reuse (extended maintenance; in millions)	Savings with extended maintenance (in millions)
1	23.80	18.45	16.00	-10.65
2	46.75	36.23	17.60	-7.08
3	106.55	82.58	19.20	4.77
4	164.05	127.14	20.80	16.11
5	203.25	148.70	38.40	16.15
6	240.40	169.13	41.60	29.67
7	305.10	204.72	44.80	55.58
8	368.95	239.83	48.00	81.12
9	486.95	304.73	51.20	131.02
10	597.45	365.51	54.40	177.54

4 Presenting Cost Comparisons in a Business Case

The NRO presented a business case for following an incremental strategy. The up-front investment costs were much lower than for a full asset base. The NRO obtained a product line demonstration capability in less than two years. The assets were not comprehensive, but were selected to address areas with the highest degree of commonality within the product line. The incremental strategy allowed the NRO to track the anticipated results of the business case and to show whether the organization was meeting the goals listed in Section 3.2. If projections do not materialize, the NRO's investment may be cut off.

Table 4 through Table 6 show the cost savings that the organization will realize from the product line investment. While total system costs decrease by 15% for Increment 1, they decrease by 30% for Increment 3. Plotting these results offers a clear illustration of the cost benefits from investing in product line assets. The trend charts offer even more dramatic results. Figure 1 plots the results from Increment 1. With the limited DOR of 25% and an assumed COR of 25%, the gap between product line costs versus stovepipe costs grows dramatically. Increments 2 and 3 illustrate the effect of decreasing the COR and increasing the DOR, respectively. During these increments the gap grows more quickly. These savings will come even sooner if more systems are produced than the nominal two systems per year.

Figure 2 looks at the three increments and their respective cost savings, and shows that the third increment generates much greater savings. However, the figure does not reflect the possible effect of lower refresh rates for the new assets. Two or three years out, the costs of reuse will increase as the software for the newly covered domains changes. Figure 2 also shows a dip when the expenses for domain expansion hit, between systems 4 and 7. The organization must be prepared to absorb this negative return, and possibly others due to the refresh rate, until the costs of new asset development are recovered during Increment 3.

In summary, the ABC approach supports analysis when an organization is considering alternative and incremental approaches for fielding a product line. The analysis offers several advantages over a straight cost comparison. This approach allows for a series of increments for introducing assets into a product line and presents ranges of savings based on the costs of using those assets. As actual cost data are collected for product development, they can be included in the cost models for better estimating and for accurate tracking of results. They may then be used by the organization to make decisions regarding investment in new product lines.

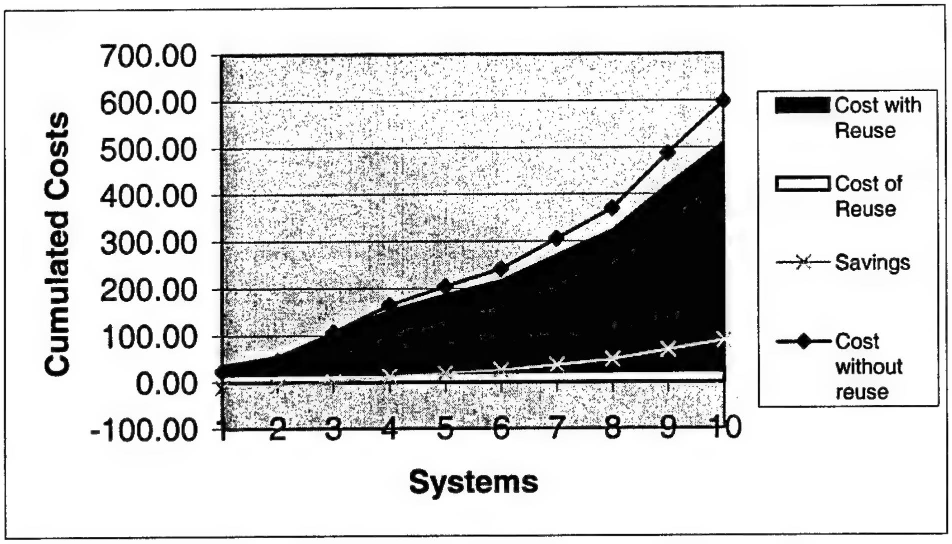


Figure 1: Increased Savings (in Millions) Under a Product Line Approach

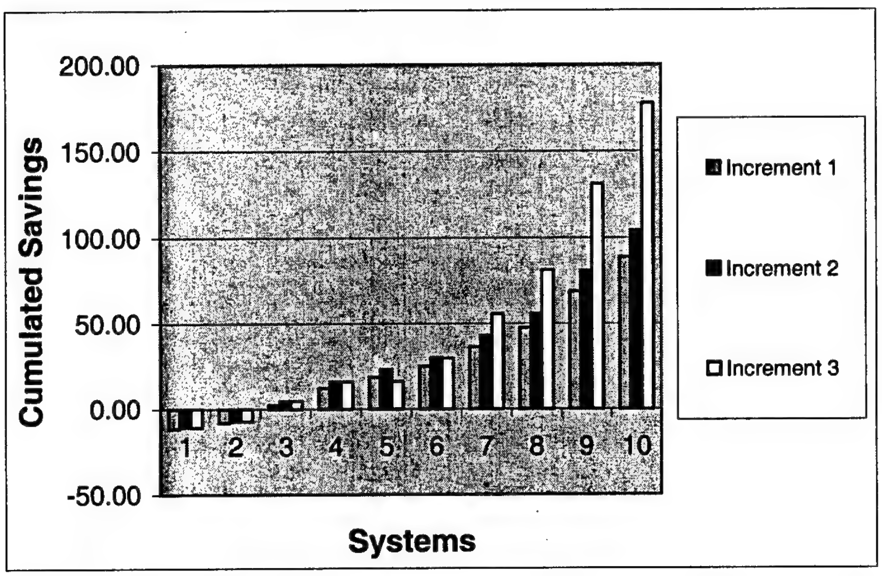


Figure 2: Comparison of Savings (in Millions) Under Each Increment

References

URLs valid as of the date of publication of this report.

[Bergey 01] Bergey, John et al. *Fourth DoD Product Line Practice Workshop Report* (CMU/SEI-2001-TR-017, ADA399205). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents/01.reports/01tr017.html>.

[Boehm 81] Boehm, Barry. *Software Engineering Economics.* Englewood Cliffs, NJ: Prentice-Hall, 1981.

[Bosch 02] Bosch, Jan. "Maturity and Evolution in Software Product Lines: Approaches, Artifacts and Organization," 257-271. *Proceedings of the Second Product Line Conference* (SPLC2). San Diego, CA, August 19-22, 2002. New York, NY: Springer-Verlag, 2002.

[Brownsword 96] Brownsword, L. & Clements, P. *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016, ADA315802). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. <http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.016.html>.

[Chastek 02] Chastek, Gary & McGregor, John D. *Guidelines for Developing a Product Line Production Plan* (CMU/SEI-2002-TR-006, ADA407772). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. <http://www.sei.cmu.edu/publications/documents/02.reports/02tr006.html>.

[Clements 01] Clements, Paul; Cohen, Sholom; Donohoe, Patrick; & Northrop, Linda. *Control Channel Toolkit: A Software Product Line Case Study* (CMU/SEI-2001-TR-030, ADA396286). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. <http://www.sei.cmu.edu/publications/documents/01.reports/01tr030.html>.

[Clements 02]	Clements, Paul & Northrop, Linda M. *Software Product Lines: Practices and Patterns*. Reading, MA: Addison-Wesley, 2002.
[Dager 00]	Dager, James C. "Cummins' Experience in Developing a Software Product Line Architecture for Real-Time Embedded Diesel Engine Controls," 23-46. *Software Product Lines: Experience and Research Direction, Proceedings of the First Software Product Line Conference* (SPLC1). Denver, CO, August 28-31, 2000. Boston, MA: Kluwer Academic Publishers, 2000.
[Glass 03]	Glass, Robert L. *Facts and Fallacies of Software Engineering.* Reading, MA: Addison-Wesley, 2003.
[Lim 98]	Lim, W. C. *Managing Software Reuse.* Upper Saddle River, NJ: Prentice-Hall, 1998.
[Poulin 97]	Poulin, J. S. *Measuring Software Reuse.* Reading, MA: Addison-Wesley, 1997.
[Reifer 97]	Reifer, D. J. *Practical Software Reuse.* New York, NY: Wiley, 1997.
[Weiss 99]	Weiss, D. & Lai, C. T. R. *Software Product-Line Engineering: A Family-Based Software Development Process.* Reading, MA: Addison-Wesley, 1999.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY
(Leave Blank)

2. REPORT DATE
July 2003

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Predicting When Product Line Investment Pays

5. FUNDING NUMBERS
F19628-00-C-0003

6. AUTHOR(S)
Sholom Cohen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU/SEI-2003-TN-017

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

A product line approach may appear very attractive, with obvious benefits in speedier time to market and higher quality, however many organizations demand financial justification before proceeding. Without knowing costs, the decision makers won't budget funds or personnel to carry out the up-front asset construction tasks. In addition, not all organizations are ready to commit up front to a full asset set, one that covers most if not all product line features. Many managers favor an incremental approach to product line adoption, one that first tackles areas of highest and most readily available commonality, earning payback early in the adoption cycle.

This report defines key factors to consider in taking an incremental approach to fielding a product line. An organization building a business case can apply these factors to show that product line investment can result in product development savings. The example presented here shows a net savings of almost $180 million in projects that would have cost about $600 million under traditional development approaches. The $180 million in savings takes into account an investment of $54 million in product line start-up costs. The example also illustrates ways to present the data needed to make a compelling business case.

14. SUBJECT TERMS
product line, product line applications, product line benefits, product line business case, product line costs, product line case studies

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL